UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/741,502	12/19/2000	Theodore S. Hills	TAJ-0001	5730

23413 7590 09/22/2004

CANTOR COLBURN, LLP
55 GRIFFIN ROAD SOUTH
BLOOMFIELD, CT 06002

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 09/22/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/741,502	HILLS, THEODORE S.
	Examiner	Art Unit
	Tuan A Vu	2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *28 June 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *32,48* is/are allowed.

6)☒ Claim(s) *1-7,10-15,17-23,26-31,33-39 and 42-47* is/are rejected.

7)☒ Claim(s) *8,9,16,24,25,40 and 41* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 6/28/2004.

As indicated in Applicant's response, claims 1-2, 5,10, 13-15,17-18, 21, 26, 29-31, 33-34, 37, 42, 45-47 have been amended. Claims 1-48 are pending in the office action.

Claim Objections

2. Claims 1, 13-15, 18, 29-31, 34, and 45-47 are objected to because of the following informalities: the limitation recited as 'wherein the describing software-visible physical objects is performed using ... classes' appears to have a grammatical incongruity and should be corrected, e.g. as in 'wherein the describing of software-visible physical objects is performed using ... classes'.

Appropriate correction is required.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

4. Claims 1, 17 are rejected under 35 U.S.C. 102(b) as being anticipated by Juergen Boldt, "Complete MOF 1.3 specification", document: *formal/00-04-03(MOF 1.3)*, date: 04/03/2000, URL: *http://www.omg.org/cgi-bin/doc?formal/00-04-03* (hereinafter Boldt).

As per claim 1, Boldt discloses a computer programming method comprising:

describing data types (e.g. ModelElement, NameSpace, GeneralizableElement, TypedElement - § 3.4.1; 3.4.2; 3.4.3; 3.4.4 – Note: meta-modeling is equivalent to describing

programming language data in computer programming) as abstract data types without default or implicit implementation and comprising only non-concrete data types (Note: no concrete implementation, i.e. memory storable variables, being implicitly created when an abstract superclass is declared as a signature – see *IDL* describing interface to implement these abstract superclasses)

distinguishing the abstract data types from classes or interfaces which are classified as abstract and concrete (see IDL: *interface TypedElementClass* pg. 3-31; *interface GeneralizableElementClass*- pg. 3-29, pg. 3-28; Classifier – pg. 3.4.5; Interface DatatypeClass pg. 3-34 – Note: class Classifier distinguisches from superclass GeneralizableElement and is abstract while interface DataTypeClass is concrete from abstract superclass Classifier); and

describing representations of values of the abstract data types as states of classes of objects with corresponding interfaces (see IDL interfaces - pg. 3-28, 3-29, 3-31, 3-34).

As per claim 17, this is storage medium and machine-readable code version of claim 1; hence is rejected using the corresponding rejections as set forth therein.

5. A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 13-15, 29-31, 33, and 45-47 are rejected under 35 U.S.C. 102(b) as being anticipated by Van Preat et al. (USPN: 5,854,929).

As per claim 13, VanPreat discloses a method of compilation comprising:

generating a description of computer architecture as a first library (e.g. *LIB format* - col. 22, lines 7-27; col. 18, lines 25-64), the description including software-visible physical objects of a computer as instances of classes, the classes comprised of methods for accessing and

manipulating said physical objects (e.g. Fig. 5, 7a, 7b; Fig. 9b; *vertices, storage elements, representing operations* - col. 5, lines 5-24; Fig. 13-15 – Note: *Memory, Register, ISG*, in Fig. 7b, 9b amount to physical objects as software-visible objects associated with methods being instantiated for design of computer objects being modeled, and simulated); wherein the describing of the software-visible physical objects is performed using OO classes and methods described as implemented by an instruction set (e.g. Fig. 1-3; *nML description* – col. 18, line 35 to col. 19, line 32; col. 22, lines 28-30; *C++* - col. 23, lines 2-28 – Note: description to specify processor instructions and assembly code inherently discloses specific description enabling selection of appropriate instruction set of target processor architecture);

implementing a record of high level objects (e.g. Fig. 6; *cdfg files* , col. 22, lines 7-27) using the description stored in the first library; and binding a source program to implementations on the record of high-level objects to produce machine instructions dependent on the computer architecture (e.g. col. 22, line 61 to col. 23, line 21).

VanPreat does not explicitly disclose that the record of high-level objects being implemented with a source program is a second library; however, Vanpreat discloses generating of ISG specification in form of a library (e.g. *Library L* - col. 11, lines 36-55) to translate data of the first library (col. 22, lines 28-55); hence has disclosed such record to be a second library.

As per claim 14, VanPreat discloses a method of re-targeting comprising the steps of generating (first description/first library); generating a second description (second library) as mentioned in claim 13 from above; these steps being therefore rejected using the same corresponding rejections as set forth therein.

Further, VanPreat discloses implementing first description using the second library to produce a third library and binding a source program to implementations on the third library (e.g. *new lib and .isg files* - col. 22, lines 53-55).

As per claim 15, VanPreat discloses a computer programming method comprising describing software-visible physical objects (e.g. Fig. 5, 7a, 7b; Fig. 9b; *vertices, storage elements, representing operations* - col. 5, lines 5-24 -Note: *Memory, Register, ISG*, in Fig. 7b, 9b amount to physical objects as software-visible objects associated with methods being instantiated for design of computer objects being modeled, and simulated); wherein the describing of the software-visible physical objects is performed using OO classes and methods described as implemented by an instruction set (e.g. Fig. 1-3; *nML description* – col. 18, line 35 to col. 19, line 32; col. 22, lines 28-30; *C++* - col. 23, lines 2-28 – Note: description to specify processor instruction set inherently discloses specific description enabling selection of appropriate instruction set)

As per claim 29, this is medium/machine-readable code version of claim 13; hence is rejected using the corresponding rejections as set forth therein.

As per claim 30, this is medium/machine-readable code version of claim 14; hence is rejected using the corresponding rejections as set forth therein.

As per claim 31, this is storage medium and machine-readable code version of claim 15 or claim 18; hence is rejected using the corresponding rejections as set forth therein.

As per claim 33, this is propagation medium/encoded signal version of claim 1; hence is rejected using the corresponding rejections as set forth therein. At the time the invention was

made, the use of propagation medium to embody a product to be disseminated across the internet was a known concept, hence this medium is implicitly disclosed.

As per claim 45, this claim includes the propagation medium signal as addressed in claim 33; and is propagation medium signal version of claim 13; hence is rejected using the corresponding rejections as set forth in claim 13.

As per claim 46, this claim includes the propagation medium signal as addressed in claim 33; and is propagation medium signal version of claim 14; hence is rejected using the corresponding rejections as set forth in claim 14.

As per claim 47, this claim is propagation medium signal version of claim 15; hence is rejected using the corresponding rejections as set forth in claim 15.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 2-7,10-12, 18-23, 26-28, 34-39, and 42-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Juergen Boldt, "Complete MOF 1.3 specification", document: *formal/00-04-03(MOF 1.3)*, date: 04/03/2000, URL: *http://www.omg.org/cgi-bin/doc?formal/00-04-03*, as applied to claims 1, 17, 33, in view of Van Preat et al., USPN: 6,477,683 (hereinafter Van Preat)

As per claim 2, Boldt discloses software-visible objects as instances of classes, the classes comprised of methods for accessing and manipulating the software-visible objects, wherein the describing of those software-visible objects is performed with object-oriented classes(e.g. § 3.9.4; 3.9.5; *add_contents, remove_contents, modify_ contents* – pg. 3-24, 3-25 – Note: Functions as *Set( ), findElementByTypeExtended(); select()* or *modify_contents()* are for accessing and manipulating software-visible objects)

But Boldt does not disclose that the software-visible objects are software-visible physical objects of a computer as instances of classes nor are those class methods are for accessing and manipulating those physical objects. The use of meta-description for modeling the behavior of target system such as in object-oriented programming for implementing and designing of hardware circuitry or computer architecture was a known concept at the time the invention was made. Van Preat, in a method using description language for enable modeling of hardware design similar to the IDL language used by Boldt, discloses nML description language in conjunction with abstract superclasses (e.g. Sub – Fig. 5) to support behaviorial description of the hardware target to built (col. 18, lines 28-51) and discloses classes of C++ for implementing methods for accessing and manipulating data representing behavior of a computer (Fig. 7b; 13-15). It would have been obvious for one of ordinary skill in the art at the time the invention was made to use the approach by Boldt for declaring super abstract classes from which to derive abstract and interface classes in implementing class methods to access and manipulate data in instances of object classes representing computer physical objects as suggested by Van Preat because the use of model to implement software and hardware real-world problems was a known concept, and according to Van Preat, by means of a model in conjunction with a meta-

description language, the different phase of code generation can be facilitated with the model description which enable simulation and deriving tiem-efficient conclusions therefrom (see Van Preat - col. 3, lines 22-64).

Nor does Boldt disclose identifying when class methods are implemented directly by a computer as instructions in an instruction set. But according to the teachings by Van Preat, and the well-known concept that selecting of an instruction set in conjunction with the target computer system to design as intended by Van Preat, this limitation would have been obvious for being suggested (e.g. Van Preat: Fig. 1-3) and thus implicitly disclosed in Van Preat.

As per claim 3, Boldt discloses multiple classes of pointer objects (e.g. § 3.5.3 – Note: interface Refersto is class pointer to be equivalent class of pointer objects)

As per claim 4, Boldt only disclose hierarchy of dependency of classes and interfaces but Van Preat discloses the tree-like organization of operations associated with class of operands in VanPreat (e.g. Fig. 3ac, 4, 5). Official notice is taken that computer operations including non-sequential actions such as event or interrupts was a known concept in the art of implementing computer architecture at the time the invention was made. In view of the attempt to control the behavioral activities and structural dependencies in a computer as taught by Van Preat, the description of what would be a computer non-sequential control operations as suggested by the Exception calls by Boldt (pg. 3-99), it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide the object-oriented C++ routines or model-derived methods by Van Preat in conjunction with the description as suggested by Boldt as to include the non-sequential control operations (exception or interrupts), such that those non-sequential type of control operations be included in the meta-description thus enabling the

simulation of their behavior according to the computer design as taught by Van Preat as set forth in claim 2, and as suggested by well-known concept as set forth from above.

As per claim 5, Boldt discloses object-oriented interfaces (see claim 1) while Van Preat discloses describing C++ programming, hence discloses at least one interfaces, classes, enumerations, subroutines as classes of objects (Note: these are implicitly disclosed in C++ or OO type of programming).

As per claims 6 and 7, Boldt does not disclose a compiler but VanPreat discloses invoking statements in a compilation

with results into a output of a compiler (col. 22, lines 61-64) and

to interpret literals in a input of a compiler (e.g. col. 19, lines 13-15; col. 22, lines 15-60; col. 23, lines 2-28); and the motivation to provide the meta modeling by Boldt in a computer design as taught by Van Preat so that a compiler is used to translate literals input with results incorporated as compiler output would have been obvious for the same benefits as set forth in claim 2.

As per claims 10 and 11, the limitation as to formal arguments in C++ or object-oriented language routines encompassing arguments such as type, reference to classes, interfaces, data attributes are implicitly disclosed in Van Preat teachings or common object-oriented programming language specification; but Van Preat does not explicitly teach specifying formal arguments as instances of objects, the object' states representing those formal arguments. Official notice is taken that passing a structure array object to describe the number of object instances being required as formal arguments was a known concept, i.e. argv[] in C++ , or suggested by the getParameterList() in Java; the state of such array object being represented as

being a NULL or non-null, and formal argument being described as object type, array pointer or *argc*, number of arguments, also all known concepts for describing specificity of such array object being passed as formal argument. Hence, the above limitation would have been implicitly disclosed in Van Preat's C++ programming.

As per claim 12, in view of the concept that a class is an argument provided in the list of formal parameters as mentioned in the rejection of claims 10-11, the teaching that a class is being parametized as part of such listing is equivalent to subroutines being parametized classes.

As per claims 18-23, these claims correspond to claims 2-7, respectively; hence are rejected with the corresponding rejections as set forth therein.

As per claims 26-28, these claims correspond to claims 10-12, respectively; hence are rejected with the corresponding rejections as set forth therein.

As per claims 34-39, these claims correspond to claims 2-7, respectively; hence are rejected with the corresponding rejections as set forth therein.

As per claims 42-44, these claims correspond to claims 10-12, respectively; hence are rejected with the corresponding rejections as set forth therein.

Allowable Subject Matter

9. Claims 8, 9, 16, 24-25, and 40-41 are objected to as being dependent upon a rejected base claim, but would be (*would be* is not the same as *will be*: emphasis added) allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. Applicant is advised that when such rewriting is effected, the claims as adjusted should be free from other form of informalities susceptible to more rejections, notably § USC 101, § USC 112 first paragraph or indefinite language.

Claims 32 and claims 48 contain allowable subject matter and *would be* allowable if fulfilling the conditions mentioned above, e.g. ensure that the claimed invention is not leading to a non-statutory subject matter, an abstract idea; or not fulfilling a practical application test.

From the above claims, the limitations referred to as deriving variable classes or variable interfaces from, respectively, a constant class or constant interface; and describing one of such variable classes/interfaces and constant classes/interfaces using a single descriptor is not disclosed or rendered obvious by teachings of any prior art.

Response to Arguments

10. Applicant's arguments filed 6/28/2004 have been fully considered but whenever applies, are not persuasive. As per the arguments in regard to which the current grounds of rejection require some response, following are the reasons as to why they are not persuasive.

(A) As per the 102(b) rejection, the Applicant has submitted that Van Preat does not teach or suggest library including 'software-visible physical objects ... comprising methods for accessing ... physical objects' (Appl. Rmrks, pg. 16, botton, pg. 17, top). The rejection has shown entities from the description tree or modeled hierarchy of objects representing methods or routine related to some physical entities (e.g. memory, register) of the computer being modeled and simulated. And the use of C++ to implement the graph in light of such analysis is equivalent to the claimed limitations as to describing those physical objects in instances of object methods to access and manipulate the data of those instances. Until the claim clearly specify what those software-visible physical objects and methods for accessing those objects consist of, a broad and reasonable interpretation has been used, thus leading to what stands in the rejection.

(B) As per the 103(a) rejections involving Van Preat, Killian, and Conner and claims 1 and 17 (Appl. Rmrks, pg. 17, bottom to pg. 20, middle), these arguments are now moot in light of the amended claims and the subsequent new grounds of rejection.

(C) As per claims 2 and 18, and the non-teaching of the instances of class whose methods are instructions implemented by the computer as instructions of an instruction set (Appl. Rmrks, pg. 20 bottom, pg. 21, top), the rejection is directed to show that Van Preat has description to specify an architecture with instructions to be compiled and implemented in C++ language to be executed in order to simulate the very computer being the target system, hence inherently discloses using the instruction set specific to that target architecture.

(D) As per claims 3 and 19, the arguments of pg. 21, 3rd paragraph are now moot in view of the new grounds of rejection.

(E) As per claims 4 and 20 (Appl. Rmrks, pg. 21,bottom) and claims 5 and 20 (Appl. Rmrks, pg. 22, 2nd para), the new grounds of rejection as a result of the changes made to claim 1 and claim 17 are now such that those arguments are moot or at least non persuasive, because it is a known concept that C++ language as suggested by Van Preat does support enumeration, subroutines, classes and non-sequential events being thrown upon exception or system fault.

(F) As per claims 6, 7, 22, and 23 (Appl. Rmrks, pg. 22, 3rd para), a broadest and reasonable interpretation has been applied when addressing the limitation recited as 'invoking statements ... input of a compiler'. Hence the rejection has shown that Van Preat has fulfilled such limitation because compiling is a process in which statements are invoked to yield a compiler output, and such invoking is for interpreting literals being submitted as input to the compiler, all that being at worst implicitly disclosed.

(G) As per claims 10 and 26 or 11 and 27 respective arguments about instances of objects, object stated, type, interface or class being formal arguments and preconditions or postconditions (Appl. Rmrks, pg. 23, 1st para). The claim 11 recites the element 'including' therefore is treated as though it only suffices that one of the elements following such 'including' is disclosed, not every single one of such listing under the ambit of 'including'; and to that effect type is disclosed in arguments list of formal parameters in C++. The arguments concerning formal arguments being described as object states or instances of objects are also fulfilled by C++ programming language as utilized by Van Preat and the explanation about an object instance being a formal argument or a state thereof being one of such parametized list is not put forth in the rejection to clarify the grounds of the rejection.

(H) As per claims 33-39, 42-46, there are no arguments that require response and concerning points raised against claims 8-9, 24-25, 40-41, these claims contain allowable subject matter and are now removed from the rejection.

Conclusion

11. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please label "PROPOSED" or "DRAFT" – please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
September 12, 2004

ANIL KHATRI
PRIMARY EXAMINER